Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 25 (2014) 230 – 237

www.elsevier.com/locate/procedia

8th International Conference on Digital Enterprise Technology - DET 2014 – "Disruptive Innovation in Manufacturing Engineering towards the 4th Industrial Revolution

Conceptual Design of a Self-Optimising Production Control System

Mittag, T. *; Gausemeier, J.; Graessler, I.; Iwanek, P.; Koechling, D; Petersen, M.

Heinz Nixdorf Institute, University of Paderborn,
Fuerstenallee 11, 33102 Paderborn, Germany

* Corresponding author. Tel.: +49 (0) 52 51 - 60 62 35; fax: +49 (0) 52 51 - 60 62 68. *E-mail address:* Tobias.Mittag@hni.uni-paderborn.de

Abstract

Current production control systems cannot react appropriately to unknown situations (e.g. the dispatch of rush jobs). They are only able to react on known situations with a predefined behaviour. In this paper the paradigm of self-optimisation will be transferred to the production control level by using a procedure model to design a self-optimising production control system. The production control is then able to react autonomously on changing operational conditions and to deduce new reaction strategies for occurring faults or disturbances. A rule based decision model is the core of the conceptual design. It is based on known and possible future faults and deducts reaction strategies. Simultaneously to them, a simulation model will be proposed, that simulates and evaluates suitable strategies.

Keywords: Conceptual Design; Self-Optimisation; Production Planning

1. Introduction

Ever shorter innovation cycles, the increasing amount of product functionality and a customisation of products lead to rising complexity of production control of current production systems [1]. However, established production control systems can only respond to familiar situations, such as certain disturbances, with a given behaviour. For this reason, the production control cannot respond adequately to unforeseen changes (e.g. cancellation of jobs) in the production process. They are not sufficiently capable of learning and accordingly only partially able to compensate disturbances in the production process or to ensure the correct dispatching of rush jobs. One solution approach to handle these challenges is the paradigm of self-optimisation. Self-optimisation describes the ability of a technical system to endogenously adapt its objective regarding changing influences and thus adapt the system's behaviour in accordance with the objectives. Behaviour adaptation may be performed by changing the parameters or the structure of the system [2]. In terms of a self-optimising production control, possible objectives are "maximising the output", "minimising the energy consumption" and "maximising the delivery reliability". Factors that affect the production control are failures of machines or missing staff, the fluctuating energy price and the current job situation. The adaption of the behaviour of the production system is conducted by a change of the structure (e.g. changing the order of the process steps) or by changing the machine parameters (e.g. variants of CNC programs). The realisation of a self-optimising production control enables permanent consideration of the current production situation and thus an optimised distribution of jobs on the machines (e.g. lathe, milling machine) at any time.

In this paper the design of a self-optimising production control is described using the specification technique CONSENS (CONceptual design Specification technique for the Engineering of mechatronic Systems) [3]. The description is structured in several interrelated aspects, e.g. environment or application scenarios. The aspects are computer-internally represented by partial models. The specification provides a holistic discipline-spanning description of a self-optimising production control [2].

doi:10.1016/j.procir.2014.10.033

2. Design of Self-Optimising Systems

2.1. Self-Optimisation

The conceivable development of communication and information technology opens up fascinating perspectives, which move far beyond current standards of mechatronics: mechatronic systems having inherent partial intelligence. We call such systems self-optimising systems. These systems adapt the priority of their objectives and behaviour autonomously in accordance with changing operating conditions.

During the operation of the self-optimising system some of its objectives may be in conflict with each other, as they cannot be pursued both to the full extend at the same time. In such cases a prioritisation of the objectives has to take place. For instance, during the adjustment of the spindle speed in a CNC turning centre the objectives ‘‘maximum feed’’ and ‘‘minimum energy consumption’’ are in conflict with each other, as energy consumption typically increases with increasing feed [2], [4].

The adjustment of objectives means that the relative weighting of the objectives is modified, new objectives are selected for pursuing or existing objectives are disregarded and no longer pursued. The adjustment of the objectives leads to an adaptation of the system behaviour. The adaptation of behaviour is realised by adjustment of system parameters and, if necessary, the structure of the self-optimising system [2].

Altogether, self-optimisation takes place as a closed-loop process, called the self-optimisation process, which consists of the three following actions [2]: (1) analysis of the current situation, (2) determination of the system objectives and (3) adjustment of the system behaviour. First, data received from other systems, the environment and the user is evaluated. Then, the fulfilment of objectives at a given time is assessed based on the results of the evaluation. Next, the system determines autonomously, which objectives it will pursue and with which priority. The loop of self-optimisation is closed by the adjustment of the system behaviour, e.g. modified allocation of work jobs to the resources.

The principle of a self-optimising production control is shown in figure 1 [2]. The information-processing unit of a self-optimising production control contains three different layers (in accordance with the Operator-Controller-Module [2], [4], [5]). The reactive layer represents the functions of a conventional production control. Regular work jobs will be allocated to the different resources and the current status will be sent back. Disturbances during the production or from the environment, like rush jobs, will be processed in the reflexive layer. The same applies to the change of constraints like the increase of energy prices. The disturbances will be matched with established potential reaction strategies. If a suitable strategy has been identified, it will be requested from the database and the reflexive layer will adapt the production. If the nature of the disturbance is unknown, the cognitive layer must deduce a new reaction strategy. The new reaction strategy will be derived by a rule based decision model before it is simulated, evaluated and ranked.

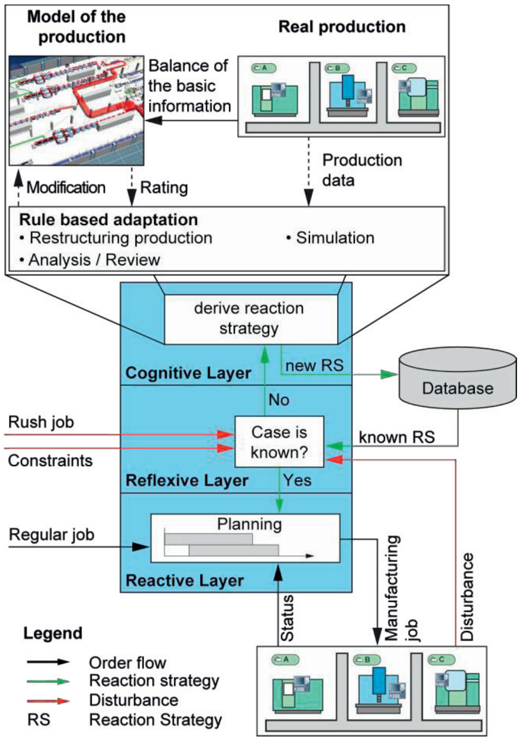

Fig. 1. Principle of a self-optimising production control

2.2. The Specification Technique CONSENS

The development of a self-optimising production control is a interdisciplinary task, as several disciplines are involved (e.g. software engineering or control engineering). There are only few design methodologies which address this issue. Most approaches focus on the respective disciplines and a holistic discipline-spanning consideration of the system is only conducted rudimentarily [2].

Especially during the early design phases, the communication and cooperation between the disciplines is necessary to establish a basis for efficient and effective system development. The approach of Model-Based Systems Engineering focuses on this aspect by means of an abstract superior system model. It enables a holistic view of the system. The system model can be specified using the specification technique CONSENS. The description of the system using CONSENS is structured into the aspects environment, application scenarios, requirements, functions, active structure, behaviour, system of objectives, shape, process sequence and resources. The aspects are computer-internally represented as partial models. The aspects are interrelated to each other and form a coherent system.

For self-optimising systems like the self-optimising production control, the aspects environment, process sequence, resources and the system of objectives are very important (figure 2).

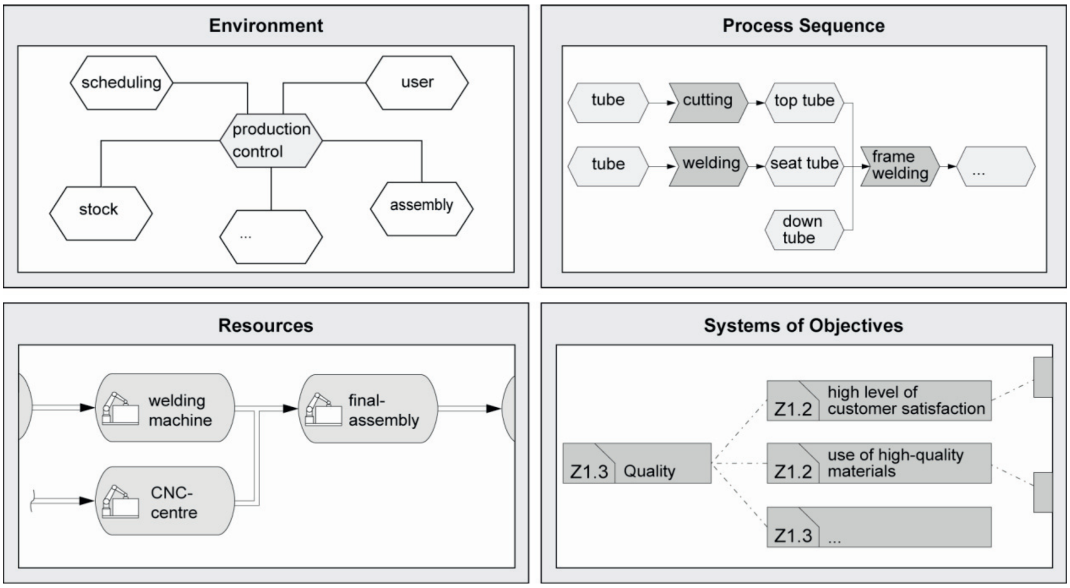

Fig. 2. Most important aspects of the self-optimising production control

The embedding of the system, which has to be developed, into its environment and the environment itself are described by the associated partial model. Relevant influences (e.g. superior systems or user inputs) will be identified and the interplay between them will be evaluated. The process sequence describes all relevant processes by a manufacturing function and attributes. The functions will be concretised into technologies and manufacturing processes during the conceptual design. Each process is characterised by at least one input object and one output object. These objects are material elements and will be described more detailed in chapter 3. The last material element is the end product of the process. Resources are necessary for the execution of the processes. They are defined e.g. as all tools, machines or personnel that are required for the process. Each process step of the process sequence is allocated to at least one resource [6]. As well as the other relevant partial model, the resources will be characterised in more detail in the following chapter.

For the self-optimising production control the objectives are very important. This aspect describes external, inherent and internal objectives of the system and their interrelations. External objectives are set from the outside of the self-optimising system; they are set by other systems or by the user. Inherent objectives reflect the design purpose of the self-optimising production control. Objectives build a hierarchy and each objective can thus be refined by sub-objectives. Inherent and external objectives that are pursued by the system at a given moment during its operation are called internal objectives. The selection of internal objectives and their prioritisation occurs continuously during the operation of the system. A detailed description of the specification technique CONSENS is provided in [2] and [3].

3. PROCEDURE MODEL

In the following section, we present a procedure model for the conception of a self-optimisation production control system. The model consists of three main phases and is shown in Figure 3. The first phase is divided into five steps that serve the purpose of the data acquisition of the current production. Furthermore, it provides the basis for the simulation model and in the following phase for the decision model.

3.1. Analysis of the Current Production

Data acquisition of the Current Production

The analysis starts with an inclusion of the current production, comprising information about the environment of the production, existing resources, processes and material elements.

The manufacturing of products is one part of the whole business operation of a company. Relevant influences (e.g. energy price) which have an effect on the production control are identified. In addition, the interdependencies between the influences are considered. A distinction is made between intended, unintended and disturbing influences. Disturbing influences on the production (e.g. rush jobs) will be classified as malfunctions or as external objectives in terms of self-optimisation. The specification of the environment is conducted according to Gausemeier et al. (chapter 2.2).

The subsequent allocation of resources is based on the interaction of processes, material elements and resources. For example, the selection of resources is limited by the size of a material element or the required manufacturing tolerance of a process [7]. Therefore, it is important to gather all the necessary information and describe them consistently.

The specification technique for the consistent description of manufacturing operations and resources is based on

CONSENS and describes the different elements with the help of specific attribute sets. Figure 4 gives an overview of the data models, in which only a part of the attributes is shown. The attributes are divided into organisational data, descriptive characteristics and rules. Non-technical information like acquisition costs or an identification number are stored as organisational data. Descriptive characteristics are used to express technical information like the dimension, the tolerance class or the technology. They are important for the selection of appropriate resources for single processes. For example, the attribute shape defines if a material element matches the size of a machine table. The rules describe the interactions between two or more attributes. For instance, a rule verifies if a resource is capable of executing a process. More information about the consistent description of manufacturing operations and resources is given in [7].

Definition of Objectives

The design purpose of the production system is expressed by its inherent objectives. The internal objectives can be derived from the results of the interaction between inherent and external objectives [8]. Thereby, the inherent objectives of a production system should be equal to the corporate objectives. Premium product manufacturer should have a quality-oriented production, whereas mass product manufacturer should rather focus on shot cycle times. Before a self-optimising production control can be implemented, the corporate objectives have to be identified and the production has to be adjusted accordingly. If several objectives are contradictory, they have to be evaluated and subsequently prioritised. The objective with the highest priority will be realised.

Prioritise Work Jobs

According to a prioritisation, the work jobs can be allocated to the resources during the later operation. This is important if a resource has a malfunction and the current work job has to be reassigned. If no alternative resource is available, the work job has to be assigned to a resource which is also able to run the required process. If this resource is already used for another work job, both work jobs have to be compared. The one with the higher prioritisation will be processed and the other will be postponed. For the prioritisation, established methods like the ABC or the XYZ analysis are used. The ABC analysis is suitable to classify material elements (products) and customers into different categories [9]. For instance, important customers or very profitable products will be assigned to class A. In addition, products are classified by the XYZ analysis into X, Y and Z categories according to their consumption rate. For example, material elements which are often sold or needed for subsequent processes will be assigned to category X and material elements which are not so important will be assigned to category Z [10]. The prioritisation is the result of performing the ABC and the XYZ method. Altogether, there are three priority levels for the work jobs

Fault Analysis

The decision model is based on universal defect classes of known and unknown disturbances or faults. Known faults can easily be identified through the evaluation of error statistics. These faults can be generalised and assigned to universal defect classes. Unknown faults can be identified by the Failure Mode and Effects Analysis. The results of the analysis can also be generalised and assigned to the defect classes.

For example, the fault: "Fail of an air valve at machining centre 1" could be generalised to "Fail at machining centre 1" and via several layers to "resource fault". Further classes are "Material faults" or "Job faults".

Development of the Simulation Model

The process control should be able to simulate new approaches in advance. For that purpose, a simulation model based on the collected data has to be developed. Any modification in the production, for instance the acquisition of new resources, must be passed into the model. Furthermore, a constant alignment of simulation model with real production is necessary. Otherwise an appropriate reaction on occurring faults is not possible. With the help of the simulation model, derived reaction strategies can be evaluated and, if necessary, optimised. The evaluation is performed for example with regard to the cycle time. In addition to that, error scenarios can be simulated in advance and suitable reaction strategies can be developed. The simulation model can be implemented with common simulation tools like Plant Simulation [11].

3.2. Design of the Decision Model

The objective of the rule-based decision model is to generate reaction strategies for the self-optimising production control. The reaction strategies are based on the internal objectives of the production system and are derived from the interaction between the inherent and external objectives. External objectives are all external influences, which include faults, rush jobs or the change of fringe condition like an increase in energy costs.

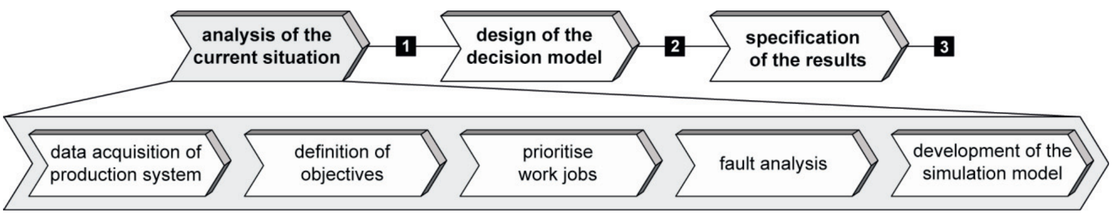

Fig. 3. Procedure model for the conception of a self-optimising production control

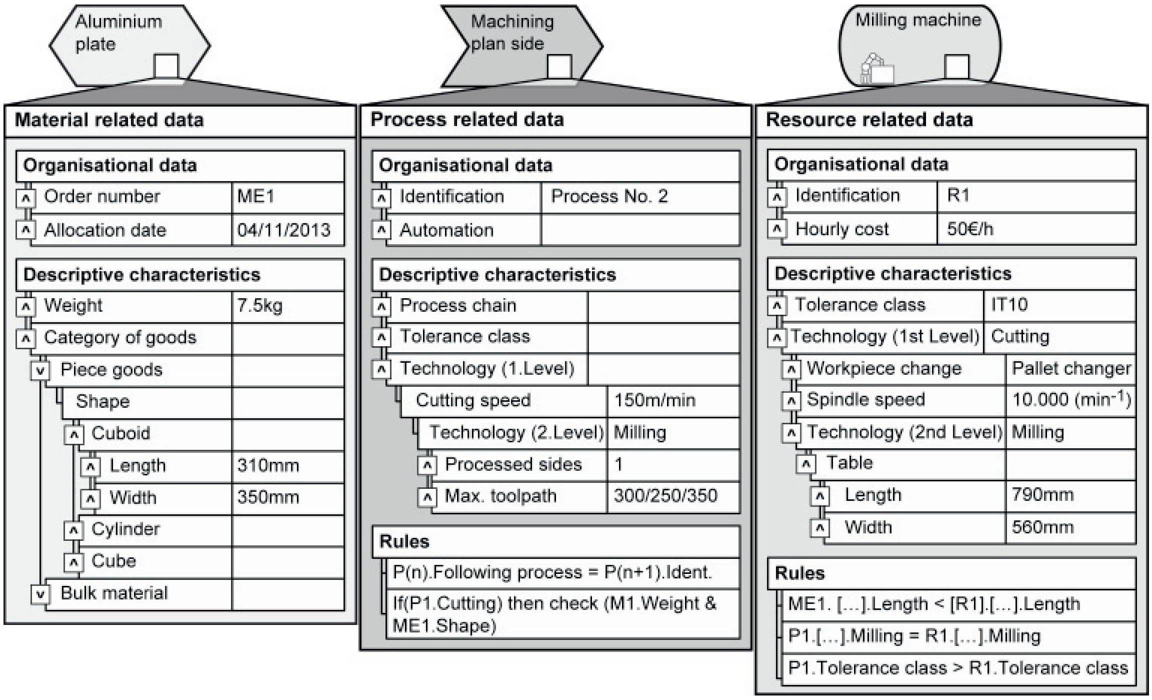

Fig. 4. Example of the data model for a material element, process and resource

The reaction strategies will be developed based on rules, which are formulated in generalised manner, so they apply for a wide range of different faults. Rules consist of premises and conclusions. The situations which have to occur before a conclusion can be drawn are described by premises. More than one premise will be combined by conjunctions like "and" (∧) or "or" (∨) [12]. A rule will be initiated if associated premises emerge at the same time. The conclusion is the assigned task for these conditions and can be a solution or a new premise which again triggers new rules. For example:

A (rush job for resource 1) ∧ B (resource 1 is (busy) → C (check for other resources)

In this case, the appearance of a rush job associated with an unavailable resource leads to the conclusion that other resources have to be checked. The rules are based on the knowledge of qualified and experienced employees, which are familiar with solving production faults. Solution processes, which have been tried and tested, will be converted into rules and the result of the process into conclusions. Subsequently, the rules and the conclusions will be abstracted to become universal. For example, the premise "resource fault" is valid for all existing resources of the company. In the following example the deduction of a reaction strategy after a resource fault is described and additionally illustrated in figure 5. A tool crack on resource 2 requires a new reaction strategy to process a pending job. The breakage of the resource leads to a tool crack on resource 2 requires a new reaction strategy to process a pending job. The breakage of the resource leads to a breakdown of resource 2. This is one of a number of predefined premises. Current work jobs for other resources or rush jobs are other premises. In figure 5, the premises, rules and conclusions are shown. The following numbered description (from I to XI) refers to the deduction of the reaction strategy. In this case, the simultaneous occurrence of the breakdown of resource" (I) and an available work job 12 (II) with the priority 2 leads to the implementation of a generally applicable rule. The rule

R()X ∧ A(.) (III)

implies that a resource is broken (R()X) and a work job occurs (A(.)) which can be applied on the existing fault (IV). The predefined conclusion C(1) (VI) will be invoked. Experts define the conclusion C(1) as: "Check alternative resources which are able to execute the process" previously. The second rule (V) implies that a rush job is pending and that the assigned resource is not available. A possible conclusion could be: "Check the prioritisation of the current work jobs". Different rules might imply the same conclusion. For example, there are no free resources for the rush job because the prioritisation of the current work job is higher than the prioritisation of the rush job this will be the case, if the current job is from an important and profitable customer and the rush job is from a smaller, not so important customer. In this instance, conclusion C(1), would be invoked again. The respective conclusion starts new paths in the decisions model because the drawn conclusions becomes a premise in the next step (VII).

If there is another resource which can execute the process (VIII) and is actually available (IX), the rule will be executed (X) and the conclusion C(2) (XI) will be assigned. The other resource could be detected by the alignment of the current resources. As described above, the allocation is based on the interaction between resources, processes and material elements. C(2) is defined as following: “allocate ‘work job 12.1’ to resource 8”. This would be a new reaction strategy for the present fault and can subsequently be simulated. It is likely

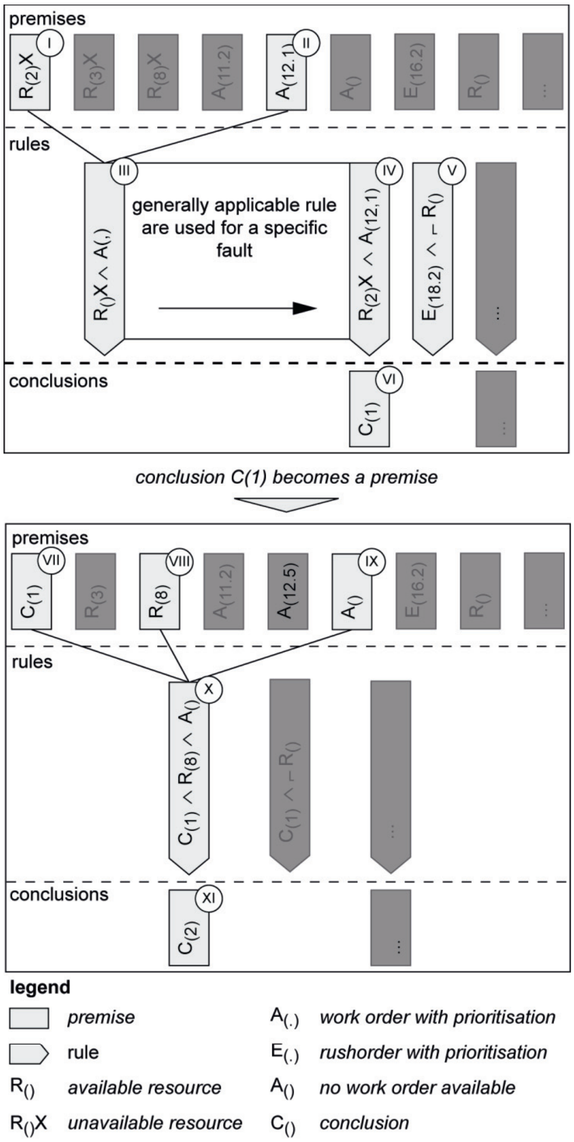

Fig. 5. Deduction of a reaction strategy

that there is more than one reaction strategy All suitable strategies are simulated and evaluated by a simulation model. The highest rated strategy is saved and made available in the future. As stated above, the evaluation is performed with regard to the cycle time.

3.3. Specification of the Results

Finally, the developed results must be specified for the software-technical implementation. This is done in accordance with the V-Model of the VDI norm 2206 [13]. There are already prototypical methods for a computer-based implementation by means of the specification technique CONSENS. These methods must be further elaborated and adjusted with regard to the self-optimising production control.

4. DEMONSTRATOR

For a comprehensive validation of the self-optimising production control, a software-based demonstrator is being developed. The demonstrator is based on a simulation model of a generic production system for bicycles and is used to simulate the different effects of the selected reaction strategies deducted by the self-optimising production control. The demonstrator uses a three tier software architecture consisting of a data tier, an application tier and a client tier. Figure 6 gives an overview of the demonstrator’s architecture and the connections between the tiers as mentioned above.

The client tier is divided into a control panel to execute user-triggered production disruptions (e.g. a resource breakdown) or changing constraints (e.g. increased energy costs) and a simulated production, which is managed by the self-optimising production control and the control panel. The simulated production is used here to represent a real world production system. The basis of the simulated production system is represented a simulation model. This model contains all the material elements, process steps and resources as well as their characteristics from the bicycle factory. The model enables the impact simulation of the production control’s reaction strategies based on a preceding production disruption or basic condition change.

A server application represents the application tier of the example and controls the whole communication between the different tiers. Furthermore the server also provides the program logic for the self-optimising production control. That means, the server receives information about production disruptions or basic condition changes and – if no suitable reaction strategy can be found in the database – the server deduces new strategies as a reaction to the current fault.

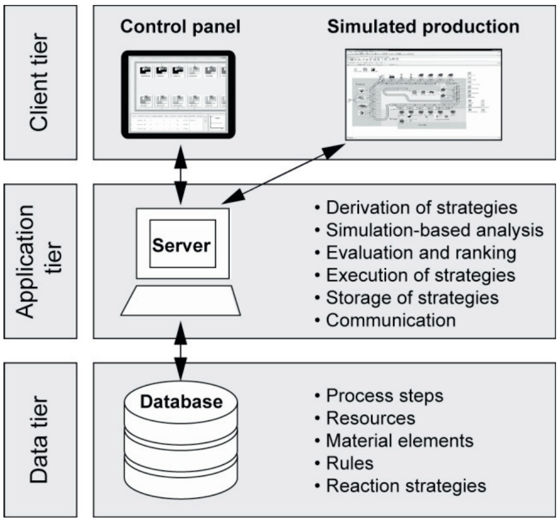

Fig. 6. Architecture of the demonstrator

For that purpose another simulation model of the production is utilised to analyse and evaluate the reaction strategies and rank the strategies based on the simulation results. Afterwards, the best reaction strategy will be executed in the simulated production of the client tier and stored into the database of the data tier.

The data tier consists of a database with a database management system and contains all required information for the self-optimising production control, e.g. the process steps or the rules.

The control panel and the server are written in the programming language Java [14] and for the database a JavaDB [15] is used. The simulated production and the simulation as part of the server are modelled in Plant Simulation [11].

5. CONCLUSION AND OUTLOOK

Non-foreseeable occurrences or unknown disturbances are problematic for current production controls. Most are unable to respond to events like a rush job or increased energy costs. In this contribution an approach has been presented that transfers the paradigm of self-optimisation to the production control system. The implementation is conducted throughout three main steps, of which the decision model forms the core. A five step analysis is the base for the decision model of the current production, which leads to a simulation model of the production. Furthermore, objectives will be defined and faults analysed. For the specification of the production we use the specification technique CONSENS. The decision model consists of universal rules and enables the production control to deduce new reaction strategies for the occurring disturbances. The prototypic implementation of the presented approach is currently under development.

In our future work we are going to extend the self-optimising production control to a superior planning level. The methods of self-optimisation should be deployed to facilitate the planning of work jobs depending on changing constraints. In addition, a combination of the production control and planning tools will be able to create a superior production control, which can coordinate work jobs between more than one production location.

Furthermore, the development of the software-technical-implemen-tation by means of the specification technique CONSENS must be extended to describe the current production in more detail. To conclude, the prototypical implementation of the self-optimising production control must be completed.

6. ACKNOWLEDGEMENT

The work in this paper is based upon investigations of the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster "Intelligent Technical Systems OstWestfalen-Lippe" (it's OWL) as well as the Collaborative Research Centre (CRC) 614, which is kindly supported by the German Research Foundation (DFG). The authors are responsible for the contents of this publication.

7. REFERENCES

[1] Brecher, C. (2011): Integrative Produktionstechnik für Hochlohnländer. Springer Verlag, Berlin.

[2] Gausemeier, J.; Rammig, F.J.; Schäfer, J. (2014): Design Methodology for Intelligent Technical Systems. Springer, Berlin.

[3] Gausemeier, J.; Lanza, G.; Lindemann, U. (2012): Produkte und Produktionssysteme integrativ konzipieren. Carl Hanser Verlag, München.

[4] Dorociak, R.; Gaukstern, T.; Gausemeier, J.; Iwanek, P.; Vaßholz, M. (2013): A methodology for the improvement of dependability of self-optimizing systems. In: Production Engineering. Volume 7, Issue 1, Springer Verlag, Berlin

[5] Strube, G. (1998): Modelling Motivation and Action Control in Cognitive Systems. In: Schmid,U.; Krems, J. F.; Wysocki, F.: Mind Modelling. Pabst, Berlin.

[6] Gausemeier, J.; Dumitrescu, R.; Kahl, S.; Nordsiek, D. (2010): Integrative development of product and production system for mechatronic prodcuts. In: Robotics and Computer-Integrated Manufacturing. Volume 27, Issue 4, p 772 - 778

[7] Rehage, G.; Bauer, F.; Gausemeier, J. (2013): Specification Technique for the Consistent Description of Manufacturing Operations and Resources. In: Proceedings of the 5th International Conference on Changeable, Agile, Reconfigurable and Virtual Production, München.

[8] Gausemeier, J.; Rammig, F.J.; Schäfer, W. (2009): Selbstoptimierende Systeme des Maschinenbaus – Definitionen, Anwendungen, Konzepte. Universität Paderborn, HNI-Verlagsschriftenreihe, Band 234.

[9] Jodelbauer, H. (2008): Produktionsoptimierung: Wertschaff-ende sowie kundenorientierte Planung und Steuerung. Springer Verlag, Berlin.

[10] Wannenwetsch, H. (2007): Integrierte Materialwirtschaft und Logistik, Springer Verlag, Berlin.

[11]www.plm.automation.siemens.com/en_gb/products/tecnomatix/plant_design/plant_simulation.shtml; 18.11.13

[12] Kurbel, K. (1992): Entwicklung und Einsatz von Experten-systemen: Eine anwendungsorientierte Einführung in wissensbasierte Systeme. Springer-Verlag, Berlin, 2. Auflage.

[13] VDI-Richtlinie 2206: Design methodology for mechatronic systems. Verein Deutscher Ingenieure, Düsseldorf, 2004

[14] www.java.com; 18.11.13

[15]www.oracle.com/technetwork/java/javadb/overview/index.html; 18.11.13